基于 Python 中 MeCab 库对日语文章进行文本分析处理实现

于瑾麟

【摘要】文本分析处理日益变成重要的课题之一，关于 jieba 中文分词的示例已有许多，但是关于日语语言分词的相关研究甚少，本文旨在介绍 Python 中 MeCab 库对日语进行分词的功能，并且给出相关案例代码，以便根据需要实现日语分词功能。

【关键词】MeCab，文本处理，Python

分词是对自然语言处理中文本分析的基础性工作，目前在国内已经存在有许多对于汉语言分词的工具，如 jieba、LTP、SnowNLP、THULAC 等，并且已经有许多关于这类工具的使用论文。但是国内关于日语分词研究的文章尚少，在日语 NLP 界当中有一些有名的开源分词系统，如 Juman,、Chasen、MeCab 等，其中 MeCab 系统是奈良先端科学技術大学院的工藤拓开发的日文分词系统，项目开发活跃，解析效率快，目前在日文 NLP 界被广泛使用。

一、 MeCab 简介

MeCab 是基于 CRF 的一个日文分词系统，代码使用 C++ 实现，基本上内嵌了 CRF++ 的代码，同时提供了多种脚本语言调用的接口(Python，Perl，Ruby 等)。整个系统的架构采用通用泛化的设计，用户可以通过配置文件定制 CRF 训练中需要使用的特征模板。

二、 文本预处理

而在日语当中存在着同一词语汉字假名混写的情况，这种现象称之为「混ぜ書き」，例如”朋友”一词，既可以写成「友達」也可以写成「友だち」。因此，如何正确的分词变得尤为重要。其次，在文中存在着诸如标点符号，语气词等对理解文本没有实际意义的词语，应当从分词结果中去除，这些词称之为停用词。去停用词可有节省存储空间，减少停用词对理解语句造成的噪音，降低文本维度。[1]

三、 MeCab 分词

(一) MeCab 对日语分词时主要包含以下几种分词输出方式：

1) 逐词分隔输出 “-Owakati” 参数

```
import MeCab
text = "今日はいい天気ですね"
mecab_tagger = MeCab.Tagger("-Owakati")
mecab_tagger.parse(text)
```

逐词分隔输出的结果是：

```
“今日 は いい 天気 です ね“
```

在这里通过 Python 内置的 split 方法对输出后的结果按照空格切割，并且去除末尾最后一个用于换行的’\n’元素。最终得到的结果是一个以逗号分隔每个词语的列表。

```
cut_list = mecab_tagger.parse(text).split(" ")[:-1]
```

得到新的输出结果是：

```
['今日', 'は', 'いい', '天気', 'です', 'ね']
```

2) 读音输出 “-Oyomi” 参数

```
import MeCab
text = "今日はいい天気ですね"
mecab_tagger = MeCab.Tagger("-Oyomi")
cut_text = mecab_tagger.parse(text)
```

读音输出的结果是：

```
"キョウハイイテンキデスネ"
```

读音使用片假名标注。

3) ChaSen 输出 “-Ochasen” 参数

```
import MeCab
text = "今日はいい天気ですね"
mecab_tagger = MeCab.Tagger("-Ochasen")
cut_text = mecab_tagger.parse(text)
```

ChaSen 输出的结果是：

```
今日	キョウ	今日	名詞-副詞可能

は	ハ	は	助詞-係助詞

いい	イイ	いい	形容詞-自立	形容詞・イイ	基本形

天気	テンキ	天気	名詞-一般

です	デス	です	助動詞	特殊・デス	基本形

ね	ネ	ね	助詞-終助詞

EOS
```

4) 全部输出 “-Odump” 参数

```
import MeCab
text = "今日はいい天気ですね"
mecab_tagger = MeCab.Tagger("-Odump")
cut_text = mecab_tagger.parse(text)
print(cut_text)
```

全部输出的结果是：

```
0 BOS BOS/EOS,*,*,*,*,*,*,*,* 0 0 0 0 0 0 2 1 0.000000 0.000000 0.000000 0

7 今日 名詞,副詞可能,*,*,*,*,今日,キョウ,キョー 0 6 1314 1314 67 2 0 1 0.000000
0.000000 0.000000 3947

20 は 助詞,係助詞,*,*,*,*,は,ハ,ワ 6 9 261 261 16 6 0 1 0.000000 0.000000
0.000000 4822

36 いい 形容詞,自立,*,*,形容詞・イイ,基本形,いい,イイ,イイ 9 15 37 37 10 6 0 1
0.000000 0.000000 0.000000 7936
```

```
49 天気 名詞,一般,*,*,*,*,天気,テンキ,テンキ 15 21 1285 1285 38 2 0 1 0.000000
0.000000 0.000000 10214

62 です 助動詞,*,*,*,特殊・デス,基本形,です,デス,デス 21 27 460 460 25 6 0 1
0.000000 0.000000 0.000000 11527

74 ね 助詞,終助詞,*,*,*,*,ね,ネ,ネ 27 30 279 279 17 6 0 1 0.000000 0.000000
0.000000 13779

78 EOS BOS/EOS,*,*,*,*,*,*,*,* 30 30 0 0 0 0 3 1 0.000000 0.000000 0.000000
11395
```

（二） 格式化 MeCab 在“-Ochasen”参数下输出的内容

在进行分析的时候我们希望能够只输出词语和它本身的词性，不需要其他无用信息，可以使用 Python 编写以下代码来实现对文本的格式化输出。

```python
import MeCab

mecab_tagger = MeCab.Tagger("-Ochasen")

def format_text(text):
    node = mecab_tagger.parse(text)
    result = []
    for i in node.splitlines()[:-1]:
        i = i.split()
        print(i[3].split('-')[0])
        result.append((i[2], i[3]))
    return result

text = "今日はいい天気ですね"
print(format_text(text))
```

在代码定义了一个名为 format_text 的函数，其中参数值为 text，使用 Python 内置的 splitlines 方法，将字符串中的换行符作为分隔符，将字符串分割成多行，并将每一行作为列表的一个元素返回。使用索引去除返回结果中末尾的 ” EOS ” 。将返回结果符合要求的第三项词语的原形以及第四项词性添加到列表当中。最后使用 print 语句将列表输出。在这里输出结果为：

```
[('今日', '名詞-副詞可能'), ('は', '助詞-係助詞'), ('いい', '形容詞-自立'), ('天気', '
名詞-一般'), ('です', '助動詞'), ('ね', '助詞-終助詞')]
```

四、 MeCab 日语分词的实际案例分析

```python
import MeCab

# 读取文本文件
with open('NEWS.txt', 'r', encoding='utf-8') as file:
    text = file.read()
    mecab_tagger = MeCab.Tagger("-Ochasen")
    result = mecab_tagger.parse(text)
    data_list = []
    for i in result.splitlines()[:-1]:
```

```
        i = i.split()
        if i[3].split('-')[0] in ['記号', '助詞', '助動詞']:
            continue
        data_list.append((i[2], i[3]))

dict_data = {}
dict_value = {}

for word, pos in data_list:
    dict_data[word] = dict_data.get(word, 0) + 1
    dict_value[pos] = dict_value.get(pos, 0) + 1

print('词性统计：')
for pos, count in sorted(dict_value.items(), key=lambda x: x[1],
reverse=True)[:10]:
    print(f'{pos}: {count}')

print('词频统计：')
for word, count in sorted(dict_data.items(), key=lambda x: x[1],
reverse=True)[:10]:
    print(f'{word}: {count}')
```

在代码头部引入 MeCab 库，使用 open 函数打开存放于目录下的 NEWS.txt 并且使用 read 函数读取其中的内容。以”-Ochasen”参数对文本文件进行处理，在排除了助词、助动词、记号等影响文本分析的内容后将符合要求的数据添加到 data_list 列表中。创建 dict_data 和 dict_value 两个字典用于存储语词和词性和对应的值，并且在遍历过程中每读取到一次就令对应的值增加。按照关键字 lambda 对字典中的键对应的值进行倒序排序后，输出前十位的值。本案例使用了一份 8KB 的文本文件进行测试，识别结果准确。以下是以目录下的 NEWS.txt 为例执行代码的结果：

```
词性统计：

名詞-一般: 265

動詞-自立: 189

名詞-サ変接続: 144

動詞-非自立: 53

名詞-数: 50

名詞-接尾-助数詞: 37

名詞-形容動詞語幹: 28

名詞-非自立-一般: 28

名詞-副詞可能: 25

名詞-接尾-一般: 24
```

```
词频统计：
する：69
いる：43
こと：22
回収：16
人：15
日：15
ある：15
市：13
製品：13
健康：12
```

五、 结束语

本文主要介绍了功能强大的日语分词工具 MeCab，文本预处理是文本分析理解的基础，直接关系到后续文本分析的准确性，文中简要介绍了 MeCab 对文本的处理模式和相关代码示例。近年来伴随着大数据以及人工智能的发展，信息量有了爆炸性的增长，如何快速且有效的获取到自己所需要的信息已经变成了一个重要的课题。通过 MeCab 对日语文本信息的分析，有望能够更好地了解日本的经济社会文化等领域。

【参考文献】

[1] 石凤贵. 基于 jieba 中文分词的中文文本语料预处理模块实现 [J]. 电脑知识与技术, 2020, 16 (14): 248-251+257.